

Path Planning of Webot Satisficing Experiment — Workspace and C-Target Construction, Off-line EER Calulation and Milestones Generation

Min Zheng

August 24, 2017

This project considers the integrated navigation and control of an unmanned ground vehicle(UGV) deployed to visually classify multiple targets in an obstacle-populated environment. This report includes the problem setup and the contruction of the \mathcal{C}_{Target} . Then the Exptected Entropy Reduction (EER) method is applied in addition to a hybrid sampling method to obtain milestones for the roadmap.

I. WORKSPACE CONSTRUCTION

The workspace is the same as the one used for active satisficing human studies. The UGV and 30 target objects are in the same environment that consists of four rooms, denoted as the region-of-interest (ROI). Similar to the human studies, there are 30 targets, denoted as $\mathcal{T} = \{\mathcal{T}_i; i = 1, 2, \dots, 30; \mathcal{T} \subset W\}$. The targets are considered as points. A two-dimensional representation from the webot environment was extracted, and the workspace is constructed in MatLab, as shown in the figure below (Figure 2).

The construction of workspace is built in an object oriented manner with classes of obstacles, targets and robot, each has properties and functionalities. Note that the table and the chairs are treated differently from other obstacles. Since their top surfaces are higher than the robot, only their legs will be considered as obstacles.

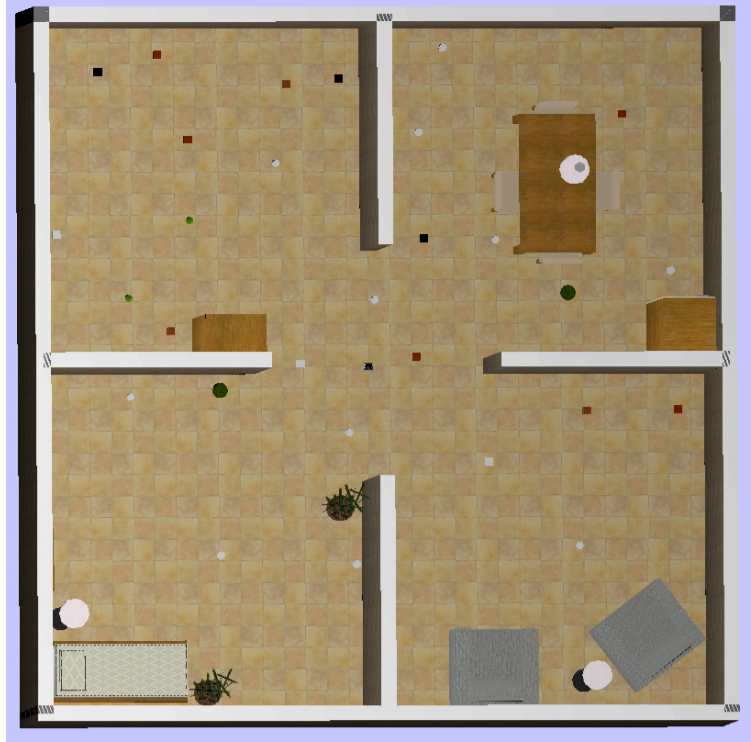


Fig. 1: Top view of the webot environment

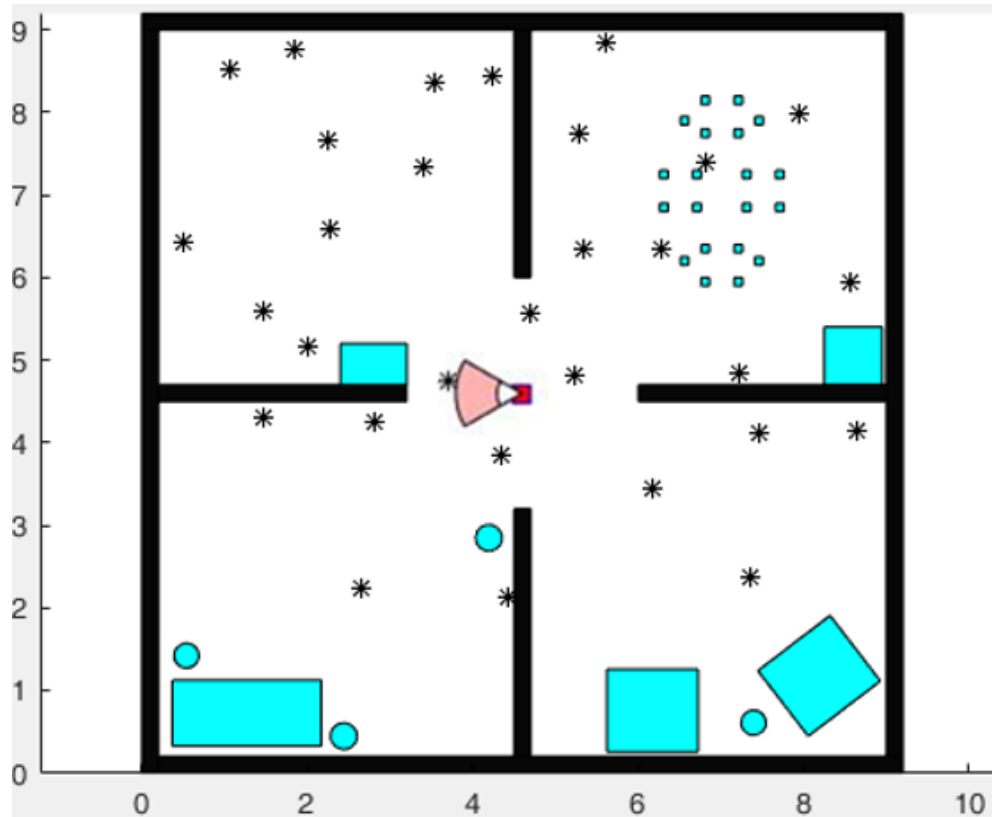


Fig. 2: 2D representation built in MatLab. Obstacles are represented by colored regions, and targets represented by the stars. The UGV is the red square, while its FOV is the red area within the fan shape.

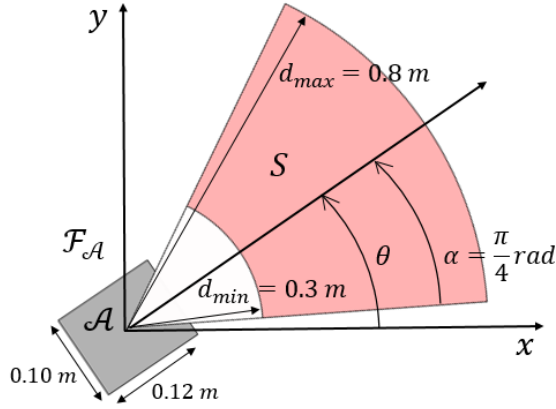


Fig. 3: Configuration and FOV of the UGV.

II. UGV'S SPECS AND FOV

The UGV sensor in the webot environment has platform geometry \mathcal{A} , and FOV geometry $S \subset \mathbb{R}^2$. The dimension of UGV is 0.12m x 0.10m in 2D. Its translation step is 0.01m, and rotation step is $\pi/12 \text{ rad}$, which are the minimum possible steps the robot can perform in translation and rotation. The UGV configuration is described in (Figure 1). The configuration vector is defined as $q \equiv [x \ y \ \theta]^T$.

In order to combine with Zeyu's visual classification project, the FOV was determined based on requirement of images processing algorithm. Successful image processing and cue classification requires the image to be taken within a minimum and a maximum distance, denoted by d_{min} and d_{max} . In human tests, the open angle of FOV for the on board camera is $\alpha = \pi/4 \text{ rad}$. Therefore the same α is used here. The images processing algorithm requires a minimum distance from the target. If the UGV is too close from the target, the camera is not able to capture a full image of the target object. Meanwhile, in the human tests software setup, the robot must be within 1m from the target to measure or classify. In such manner, the range of distance from target is conservatively set within $d_{min} = 0.3 \text{ m}$ and $d_{max} = 0.8 \text{ m}$. In order to better work with the image processing algorithm, these parameters, especially d_{max} may be subject to changes due to the highly sensitive manner of the algorithm. For example, it may be affected by whether the algorithm allows multiple objects to appear in the view.

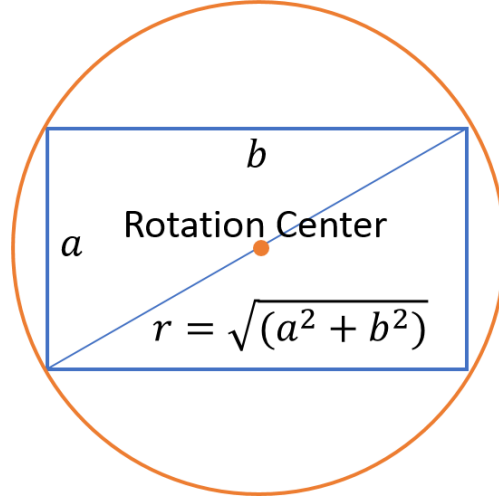
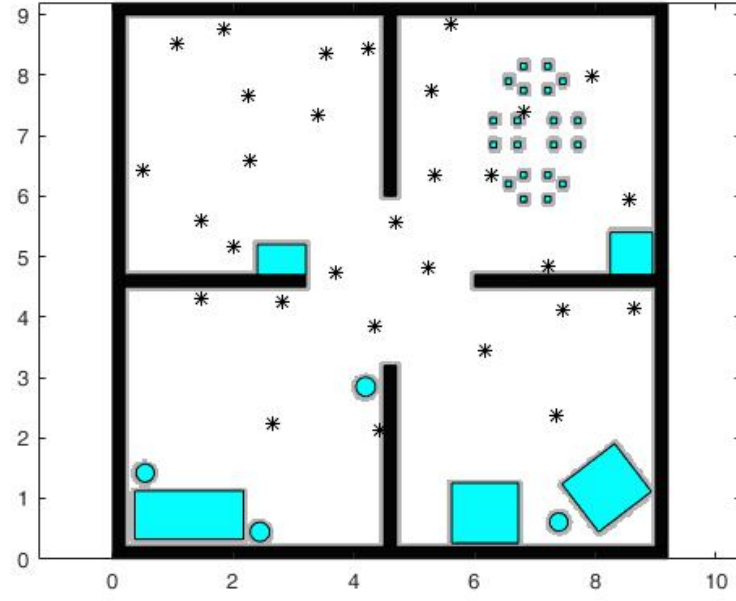


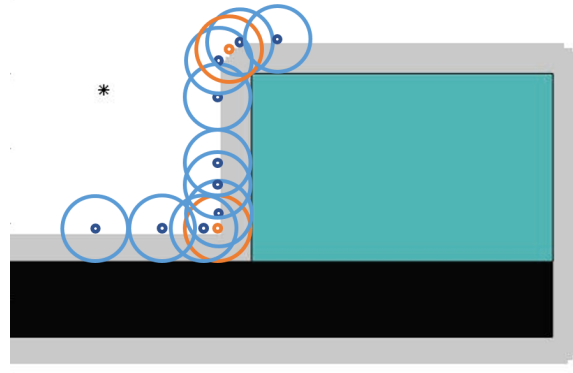
Fig. 4: The area (orange circle) swept by rotating \mathcal{A} (blue rectangle) around its center.

III. FREE CONFIGURATION SPACE, \mathcal{C}_{free}

Assume that the UGV can freely rotate. Thus $\alpha \in (-\pi, \pi)$. In x-y plane, a location is defined to be in \mathcal{C}_{free} if the UGV can freely rotate around its center, which is at the given location. The locations where some subsets of all possible α are allowed is discarded from \mathcal{C}_{free} , considering the problem of getting stuck. Given the geometry of the UGV, it requires that the bounding circle of the UGV (Figure 3) does not intersect with any obstacles. The result \mathcal{C}_{free} is shown in Figure 5.



(a) \mathcal{C}_{free} for the whole world map



(b) zoomed-in demonstration of \mathcal{C}_{free} , where the center of the circle represents the UGV center location, so that the

Fig. 5: \mathcal{C}_{free} of the given world map denoted by white region. Grey area is not in \mathcal{C}_{free}

IV. MAPPING TARGETS INTO ROBOT CONFIGURATION SPACE

Assume that the UGV camera is mounted at the center the UGV and always face the same direction as the UGV. \mathcal{C}_{Target} is defined so that the target \mathcal{T}_i in \mathcal{W} maps in the robot's configuration space \mathcal{C}_{free} to the C-target region $\mathcal{CT}_i = \{q \in \mathcal{C}_{free} | \mathcal{S}(q) \cap \mathcal{T}_i \neq \emptyset\}$. A simple $x - y$ plane demonstration (Figure 6) demonstrates the effect of the distance to the target. In (a), the robot is within \mathcal{C}_{Target} , since the target is in the robot's FOV when it faces $-\pi/2$ rad. In (b), the robot is not in \mathcal{C}_{Target} , and it can not detect the target no matter how it rotates at its current position.

Line-of-sight visibility is taken into account as well, as obstacles in the line of sight may cause occlusions with respect to the target of interest. As shown in Figure 7, let the camera state \mathbf{x} demote the position of FOV apex (\mathcal{O}_A) in inertial frame \mathcal{F}_W , and the target state \mathbf{x}_T demote the inertial position of the point target \mathcal{T} . The relative position of \mathcal{T} with respect to the camera \mathcal{F}_A is $\mathbf{r}_T = (\mathbf{x}_T - \mathbf{x})$. Then, \mathbf{x}_T thus also the target \mathcal{T} is the in the line of sight of the camera at \mathbf{x} if there are no points in the obstacle region \mathcal{B} that are co-directional with \mathbf{r}_T and closer to \mathbf{x} than \mathbf{x}_T . For implementation, the test checks if \mathbf{r}_T intersects with \mathcal{B} . An example in the workspace is shown in Figure 8.

The algorithm iterates through every configuration in \mathcal{C}_{free} . At each configuration, it checks if there exists any target in the FOV that passes the line-of-sight test. If there exists such target, this configuration is in \mathcal{C}_{Target} . A zoom-in example for the 3-D configuration is shown in (Figure 9). Then the \mathcal{C}_{Target} for the workspace is shown (Figure 10 and Figure 11).

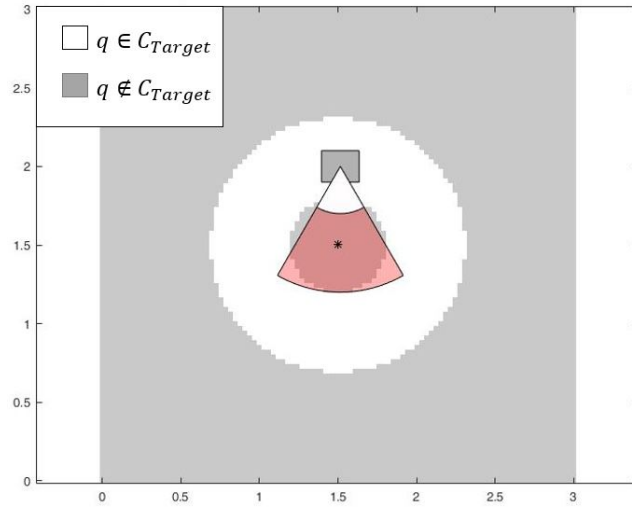
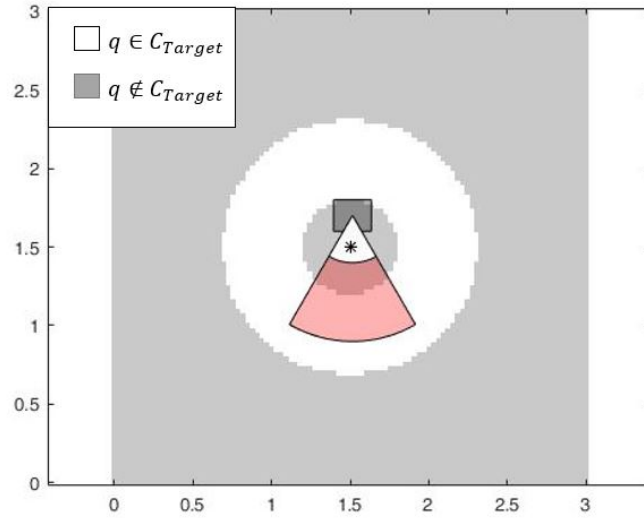
(a) $q \in \mathcal{C}_{Target}$ (b) $q \notin \mathcal{C}_{Target}$

Fig. 6: Example of C-target with one target. C-target denoted by white area, and the star denotes the target. The target must fall in the red region (robot FOV) to be classified or measured. In both examples, the robot is facing $\theta = -\pi/2 \text{ rad}$

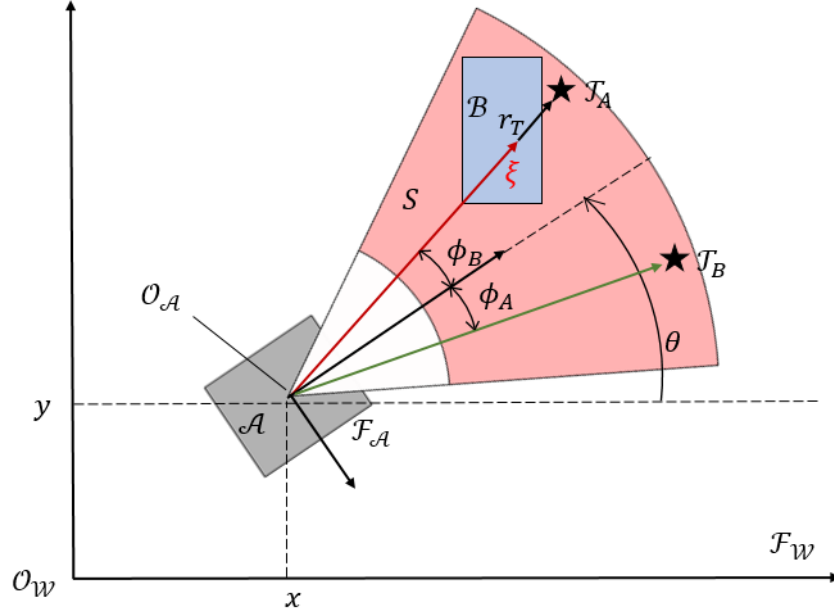


Fig. 7: With the FOV, S introduced in Figure 3, a line-of-sight visibility test is demonstrated for two targets \mathcal{T}_A and \mathcal{T}_B in S . \mathcal{T}_A is occluded by obstacle \mathcal{B} because there exists at least a point, ξ , in \mathcal{B} that fails the line-of-sight test. And \mathcal{T}_B is not occluded by obstacles since there is no such ξ .

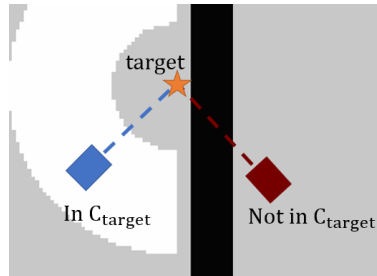
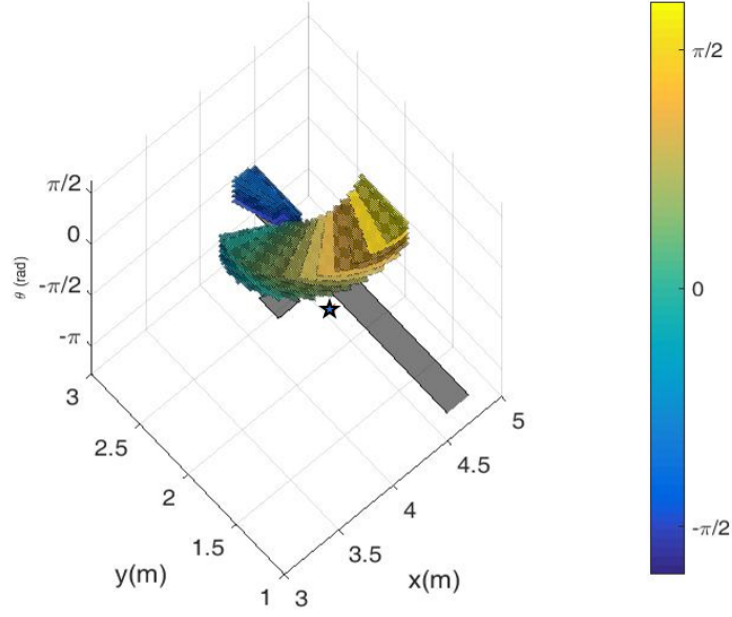
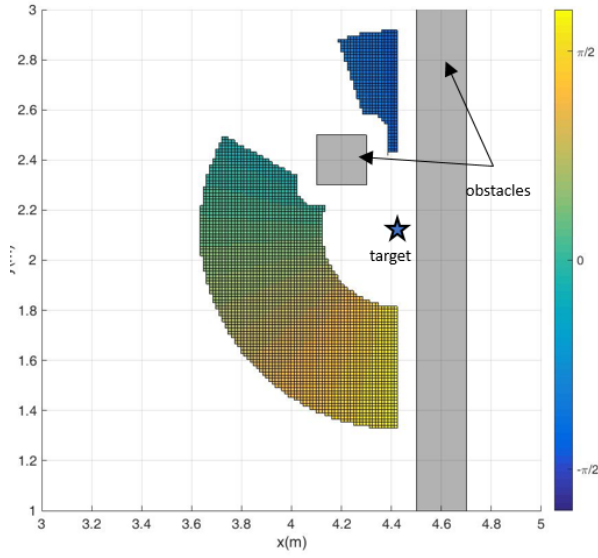
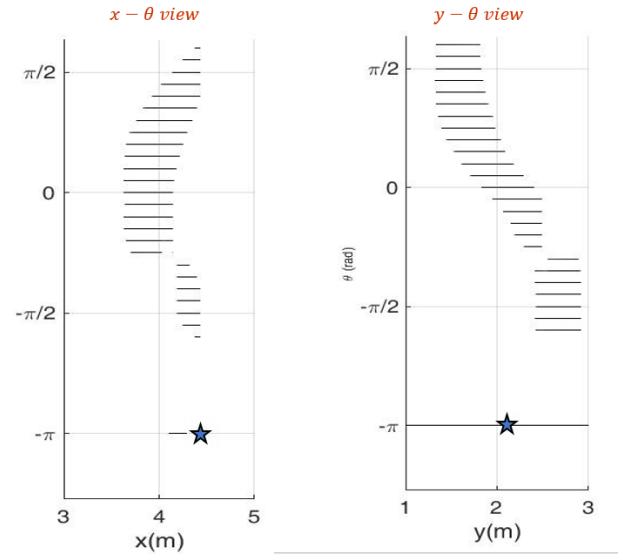
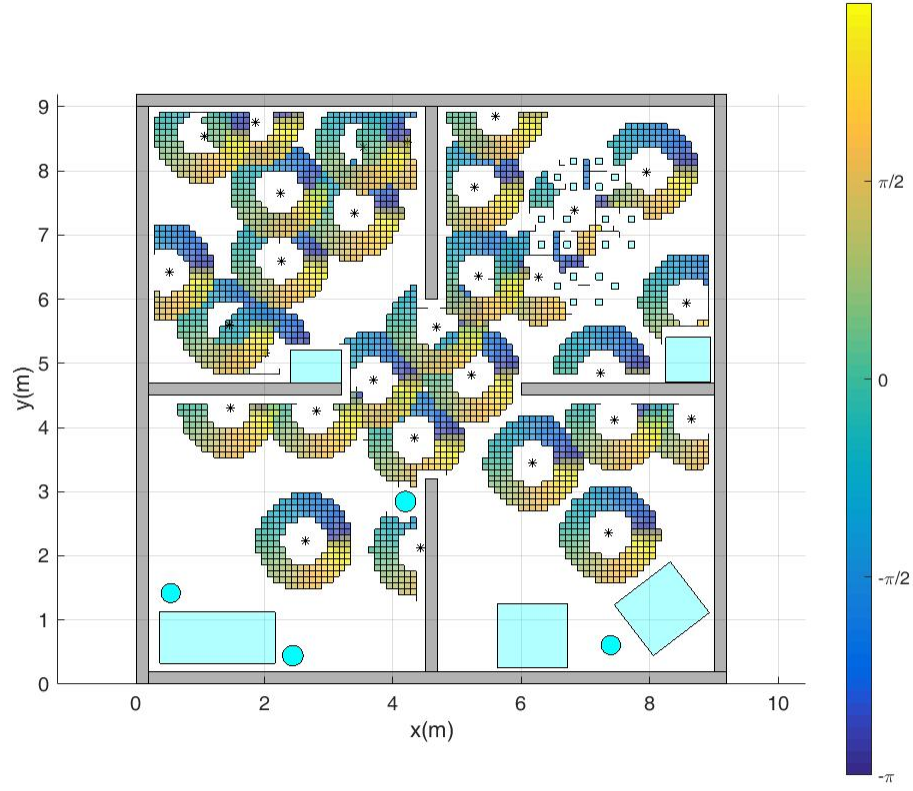
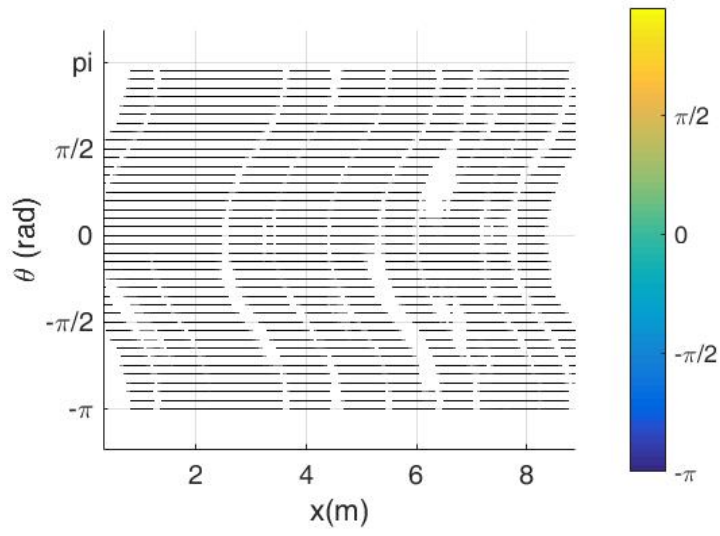


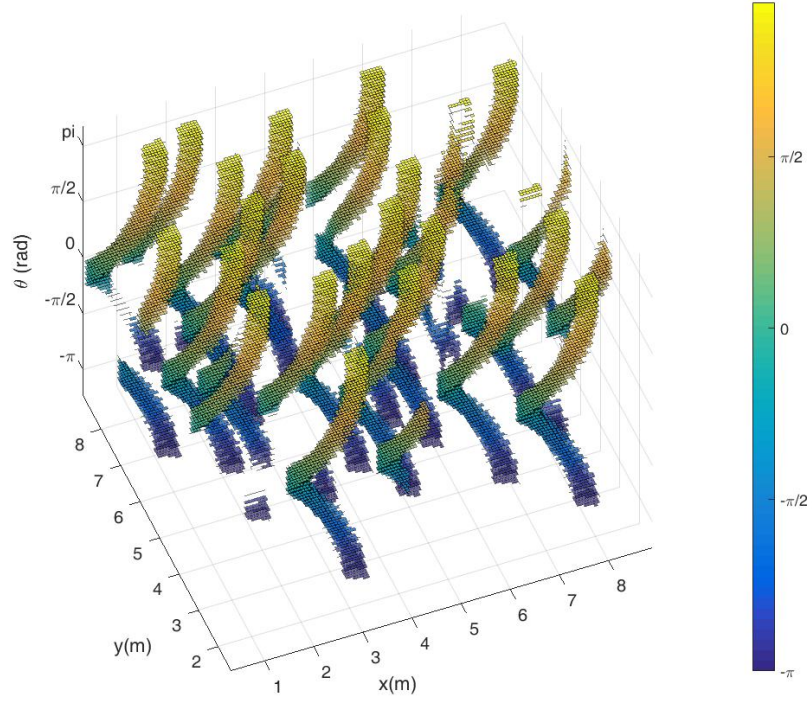
Fig. 8: Example of visibility problem handling for a target near the wall. The blue and red rectangles denote two possible UGV configurations. The dashed line denotes the robot forward direction. The blue configuration is in C_{target} , but the red is not since its view is blocked by the obstacle.



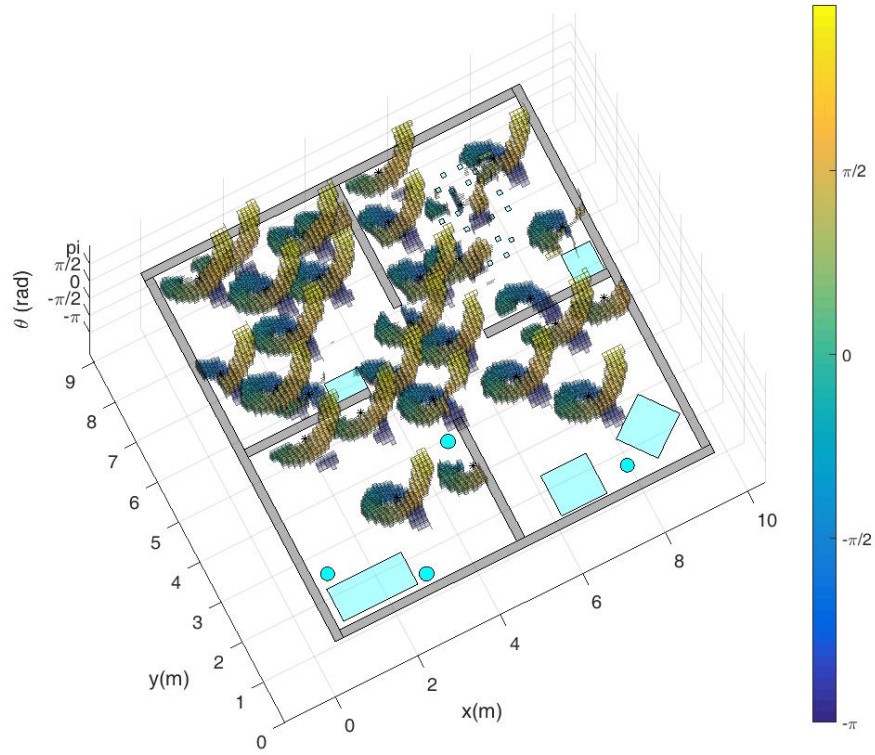
(a) perspective view

(b) $x - y$ view(c) $x - \theta$ view and $y - \theta$ viewFig. 9: An example zoom-in of \mathcal{C}_{target}

(a) $x - y$ view(b) $x - \theta$ viewFig. 10: \mathcal{C}_{target}



(a)



(b)

V. EXPECTED ENTROPY REDUCTION

EER method is applied in the milestone generation and the on-line path planning, so that the UGV finds and interacts with the targets motivated by the entropy reduction of revealing a possible feature. The target features, Bayesian Network(BN) structure and Conditional Probability Table(CPT) are the same as the one used in human tests as well as visual classification. The targets have 3 levels of features: shape, color and textures, each is a random variable named X_1 , X_2 and X_3 . By the EER calculation(in Report 170724), given the CPTs shown in Table I, the corresponding EER is shown in Figure 12 and Figure 13.

TABLE I: An example of CPT of the target objects

Cue	P(cue)							
x_{k1}	0.43 (sphere)				0.57 (box)			
x_{k2}	0.64		0.36		0.69		0.31	
x_{k3}	0.69	0.31	0.37	0.63	0.08	0.92	0.21	0.79
ID	Apple	Watermelon	Orange	Basketball	Cardboard box	Wooden box	Computer	Book
Z	0.9	0.1	0.9	0.9	0.9	0.9	0.9	0.9

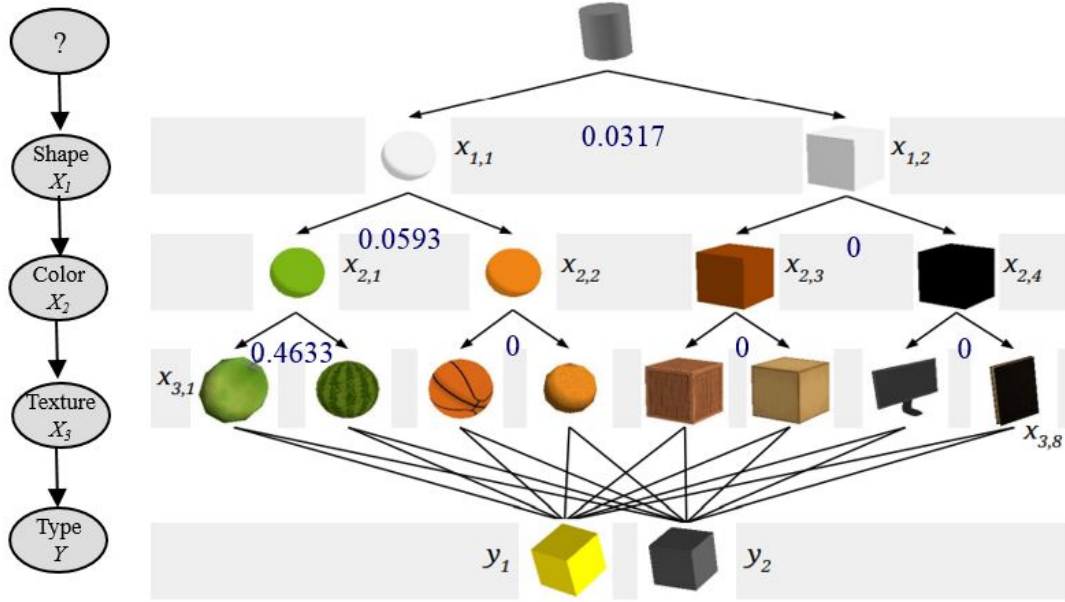


Fig. 12: Single-step EER

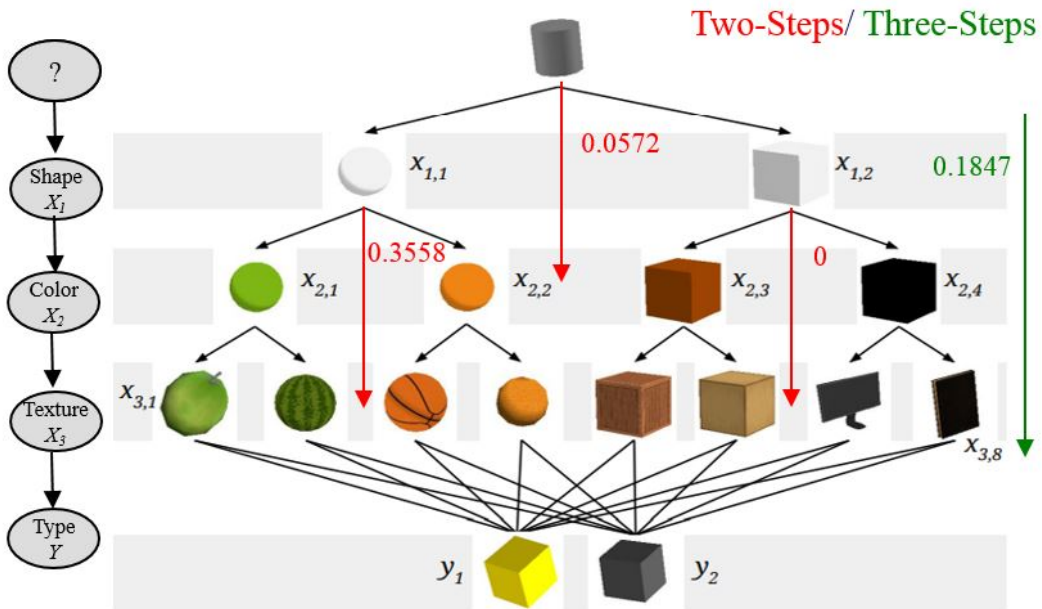


Fig. 13: Multiple-steps EER

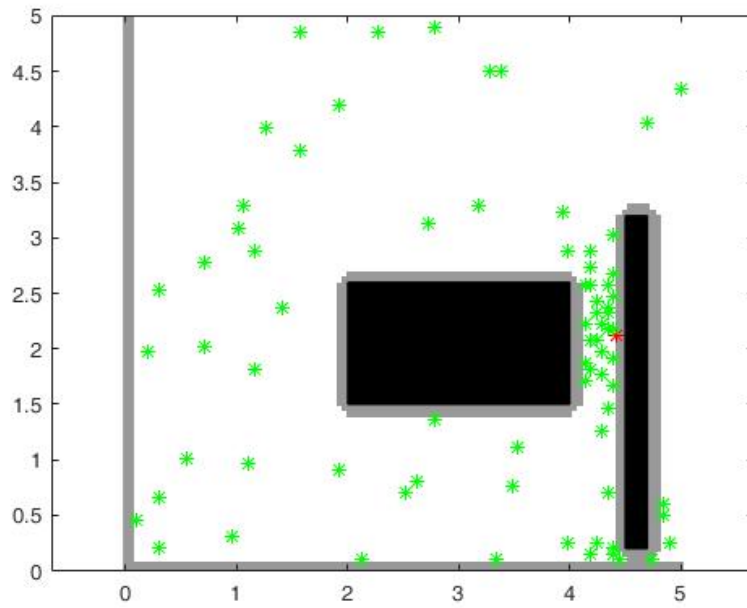
VI. MILESTONES SAMPLING

Given the resolution required for the roadmap, the time complexity and space complexity of Sampling Distribution Generation Algorithm are high. Therefore a simplified sampling method is proposed (Algorithm 1). A hybrid strategy is used to combine a bridge test for narrow passages and a uniform probability density function in wide-open regions in order to provide coverage in both. Let b and b' be two random variables representing the two endpoints of a bridge in C_{2D} . While C_{2D} is configuration in 2D space, $q \equiv [x \ y]^T$.

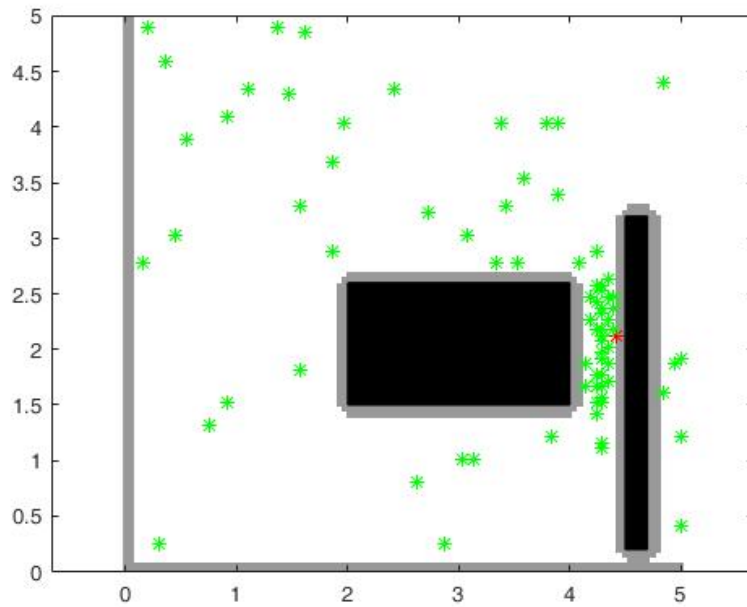
Since bridges must connect $C_{obstacles}$, b is sampled from a uniform distribution $f(b)$ over CB . And b' is sampled from a multivariate Gaussian PDF that is radially symmetric, with mean at b 's location, and a self-defined standard deviation.

Let N_H denote the total number of milestones generated using hybrid strategy. We assign a user-defined parameter $0 \leq v \leq 1$, which emphasizes either either distribution in the mixture. So that the N_H milestones should be split into $N_U = N_H * v$ and $N_{RBB} = N_H * (1 - v)$, representing uniform and RBB (Randomized Bridge Builder).

As discussed in the Narrow Passage Sampling paper, RBB may generate some milestones near the corner, which is unhelpful. To reduce the false positives near corner, the orthogonal test is performed for a RBB generated milestone. The test draws a line segment orthogonal to the bridge. The line segment length is a pre-defined variable. The milestone is accepted if the line segment is collision-free with the CB . Figure 14 and Figure 16 demonstrates an example case of using the orthogonal test.



(a) Example case, the orthogonal test is not used. Some milestones are generated near the corner, which is not helpful.



(b) Example case of applying the orthogonal test

Fig. 14: An example case on the orthogonal test

Algorithm 1 Simplified Hybrid Sampling Combined with EER

```

1: function MILESTONES( $C, C_{free}$ ) ▷ Where MS - milestone
2:   Let  $MS_U$  and  $MS_{RBB}$  be empty arrays
3:   while  $N_U$  not reached or  $N_{RBB}$  not reached do
4:     Uniformly generate a point  $q_1$  in  $C_{2D}$ 
5:     if  $q_1 \in C_{free}$  and  $N_U$  not reached then
6:       Record  $q_1$  in  $MS_U$ 
7:     else
8:       while  $q_p$  is not a valid RBB milestone do
9:         Generate  $q_2$  by multivariate Gaussian PDF
10:        if  $q_2 \in CB$  then
11:           $q_p = (q_1 + q_2)/2$ 
12:          if  $q_p \in C_{free}$  and Orthogonal Test Passes then
13:            Record  $q_p$  in  $MS_{RBB}$ 
14:          end if
15:        end if
16:      end while
17:    end if
18:  end while
19:  for every  $q$  in  $MS_U$  and  $MS_{RBB}$  do
20:    Calculate  $EER(q)$ 
21:  end for
22: end function

```

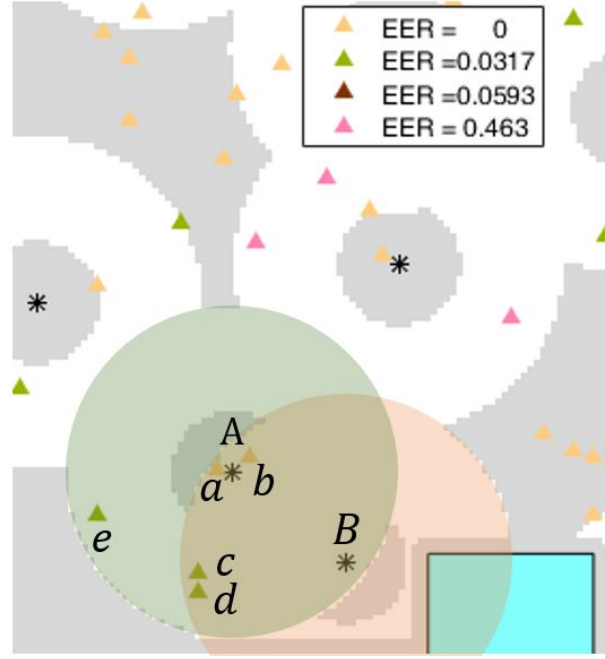


Fig. 15: Zoom in of an example of milestones, where the green and orange circles represent the regions that the target A and B can be observed correspondingly. a-e are the generated milestones.

When calculating the EER of a given target, if the configuration is in \mathcal{C}_{Target} , its EER is the single-step EER of the nearest target if there is cue unrevealed. Currently, only single-step EER is considered, because multiple-step EER is rather complex. For example, if two targets are both in the FOV, and the nearest one has only one cue to be revealed, the robot may be able to reveal cues of the second target without changing configuration. A zoom-in example with target A and B (Figure 15) demonstrates a case with the proposed protocol. Even though milestone a and b are near A, they are not in the \mathcal{C}_{Target} region of target A. Therefore their EER is the EER of target B. Milestone c and d are in the \mathcal{C}_{Target} of both A and B. Since they are nearer to target A, their EER is that of A.

The milestone generated for the entire road-map is shown in 16. A possible plan is to select, for example 50 milestones that have the highest EER. Then another 50 milestones is randomly selected from the rest. Meanwhile, the given the 2D configuration of a milestone, the third dimension θ can be determined either from reading the pre-calculated result or re-calculating by the method in Section \mathcal{C}_{Target} .

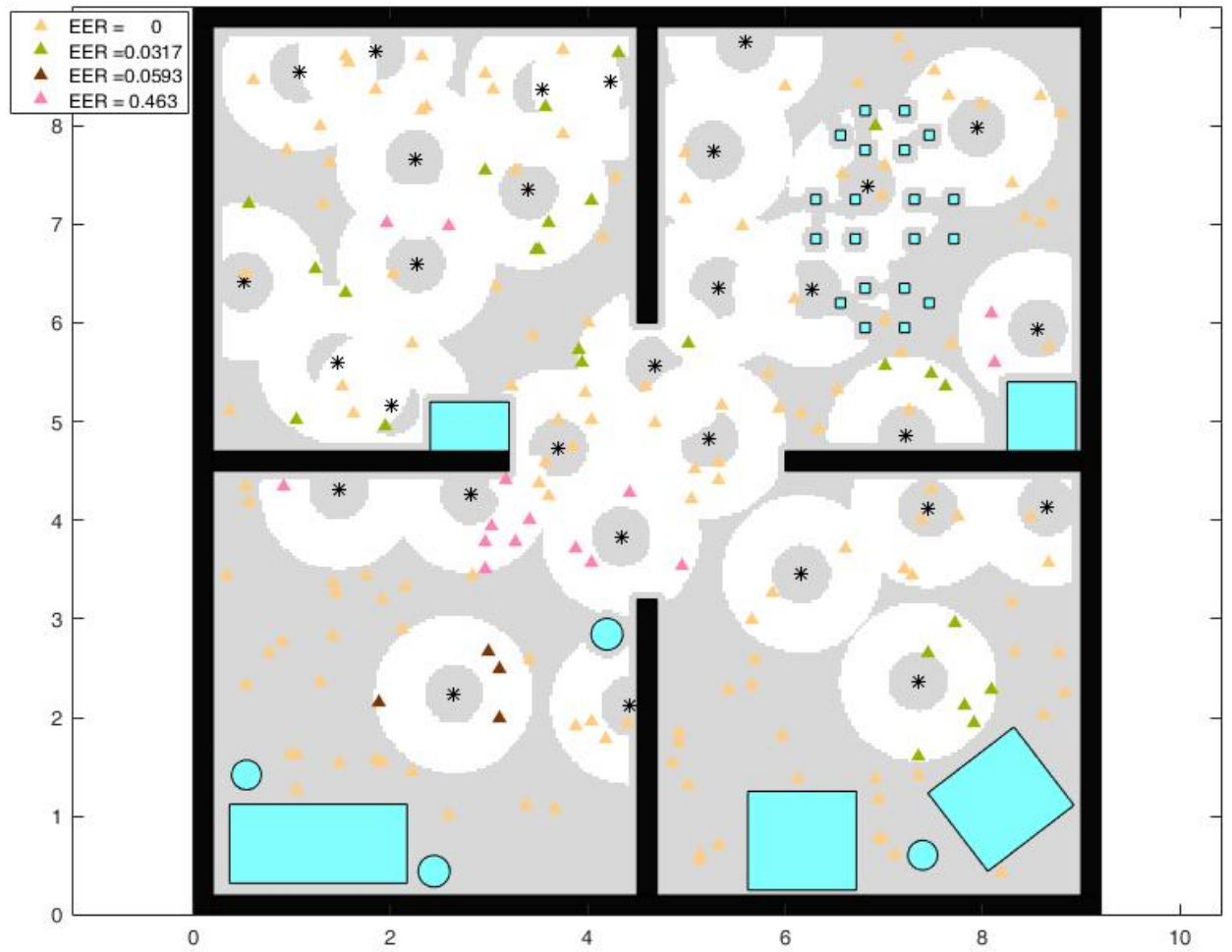


Fig. 16: Example of sampling 180 milestones